UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/598,436	06/22/2000	Charles Robert Moore	AT9-99-453	5581

42640 7590 07/13/2004

DILLON & YUDELL LLP
8911 NORTH CAPITAL OF TEXAS HWY
SUITE 2110
AUSTIN, TX 78759

EXAMINER	
HUISMAN, DAVID J	

ART UNIT	PAPER NUMBER
2183	

DATE MAILED: 07/13/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/598,436	MOORE, CHARLES ROBERT	
	Examiner	Art Unit	
	David J. Huisman	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *30 April 2004*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *17-30* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *17-21,23-27,29 and 30* is/are rejected.

7)☒ Claim(s) *22 and 28* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *22 June 2000* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 17-30 have been examined.

Papers Submitted

2. It is hereby acknowledged that the following papers have been received and placed of record in the file: Amendment and RCE as received on 4/30/2004.

Specification

3. The title of the invention is not descriptive. A new title is required that is clearly indicative of the invention to which the claims are directed. The examiner recommends incorporating the idea of the bifurcation being performed dynamically (or at run-time).

Drawings

4. The drawings are objected to under 37 CFR 1.83(a). The drawings must show every feature of the invention specified in the claims. Therefore, the portion of claims 17 and 18 involving calculating a speculative target memory address utilizing contents of at least one register **without regard** for whether said contents will be modified between calculation of the address and performing the register operation must be shown or the feature(s) canceled from the claim(s). Perhaps a flowchart can be added which includes this feature. No new matter should be entered.

Corrected drawing sheets are required in reply to the Office action to avoid abandonment of the application. Any amended replacement drawing sheet should include all of the figures

appearing on the immediate prior version of the sheet, even if only one figure is being amended. The figure or figure number of an amended drawing should not be labeled as "amended." If a drawing figure is to be canceled, the appropriate figure must be removed from the replacement sheet, and where necessary, the remaining figures must be renumbered and appropriate changes made to the brief description of the several views of the drawings for consistency. Additional replacement sheets may be necessary to show the renumbering of the remaining figures. The replacement sheet(s) should be labeled "Replacement Sheet" in the page header (as per 37 CFR 1.84(c)) so as not to obstruct any portion of the drawing figures. If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

Response to Arguments

5. Applicant's arguments, filed on April 30, 2004, with respect to the rejection(s) of claim(s) 17 and 18 under Eickemeyer have been fully considered and are persuasive. Therefore, the rejection has been withdrawn. However, upon further consideration, a new ground(s) of rejection is made in view of Chen (see below).

Claim Rejections - 35 USC § 112

6. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

7. Claim 18 recites the limitation "said plurality of registers" in the last line. There is insufficient antecedent basis for this limitation in the claim. The examiner recommends modifying the preamble to read as the preamble of cancelled claim 10.

Claim Rejections - 35 USC § 102

8. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

9. Claims 17-19, 23, 25, and 29 are rejected under 35 U.S.C. 102(b) as being anticipated by Chen and Baer, "Effective Hardware-Based Data Prefetching for High-Performance Processors," 1993 (herein referred to as Chen).

10. Referring to claim 17, Chen has taught a processor comprising:

a) a plurality of registers. See Fig. 1 and note that the instructions access registers (r2, r3, etc.).

b) instruction processing circuitry that fetches an instruction sequence for execution, said instruction sequence including a load instruction and a preceding instruction that precedes said load instruction in program order, said instruction processing circuitry, after fetching said instruction sequence for execution and prior to dispatching said load instruction for execution and responsive to detecting said load instruction within said fetched instruction sequence, translates said load instruction into separately executable prefetch and register operations. Firstly, it is inherent that processors fetch sequences of instruction for execution, wherein every instruction (except the first) has a preceding instruction. In addition, see the section on Basic

Reference Prediction on pages 612-614. It is disclosed that when a load is encountered through fetching according to the PC (program counter), a prefetch will be generated in certain instances (see bottom of page 613, column 1). More specifically, this scheme is concerned with executing multiple iterations of a loop that happens to include a load instruction. This load will execute X times, since it is in a loop which executes X times. From the bottom of page 612, column 1, it should be realized that a given load instruction (in a loop) will result in multiple prefetch operations and multiple register operations being performed. The point is to prefetch data for the load so that it is available when the actual load (register operation) is executed.

c) execution circuitry that performs at least said prefetch operation out-of-order with respect to said preceding instruction to prefetch data and subsequently separately executes said register operation to place said data into a register among said plurality of registers specified by said load instruction. Clearly, the reason that data is prefetched is so that it is available when the actual load (register operation) is executed.

d) said execution circuitry performs said prefetch operation by calculating a speculative target memory address utilizing contents of at least one register without regard for whether said contents will be modified between calculation of said speculative target memory address and performing said register operation and by thereafter initiating a fetch of said data from a memory location associated within said speculative target memory address. It should be noted from Fig.2(a) that the target prefetch addresses are calculated using at least the prev_addr field of the RPT entry. However, the data stored in the prev_addr field within the RPT comes from at least one register. For instance, looking at Fig.1, load instructions which reference registers appear at addresses 500, 504, and 512. Taking the load at address 500, 0 would be added to the value in r2

(as is known in the art) and data would be loaded from the resulting address into r4. In this case, the value in r2 is 50000, as shown in the prev_addr RPT field in Fig.3(b). The prefetch addresses are all built of the prev_addr field, as shown in Fig.2, and so the contents of the register are utilized in calculating a speculative target memory address. Finally, the examiner has found no portion of Chen which suggests calculating a speculative target memory address utilizing contents of at least one register **with regard** for whether said contents will be modified between calculation of said speculative target memory address and performing said register operation. Therefore, the examiner concludes that Chen has taught calculating a speculative target memory address utilizing contents of at least one register **without regard** for whether said contents will be modified between calculation of said speculative target memory address and performing said register operation, which meets applicant's claim limitation.

11. Referring to claim 18, Chen has taught a method, said method comprising:

a) fetching an instruction sequence for execution, said instruction sequence including a load instruction and a preceding instruction that precedes said load instruction in program order. It is inherent that processors fetch sequences of instruction for execution, wherein every instruction (except the first) has a preceding instruction.

b) in response to fetching said instruction sequence for execution and prior to execution of said load instruction, instruction processing circuitry detecting said load instruction within said fetched instruction sequence and translating said load instruction into separately executable prefetch and register operations. See the section on Basic Reference Prediction on pages 612-614. It is disclosed that when a load is encountered through fetching according to the PC (program counter), a prefetch will be generated in certain instances (see bottom of page 613,

column 1). More specifically, this scheme is concerned with executing multiple iterations of a loop that happens to include a load instruction. This load will execute X times, since it is in a loop which executes X times. From the bottom of page 612, column 1, it should be realized that a given load instruction (in a loop) will result in multiple prefetch operations and multiple register operations being performed. The point is to prefetch data for the load so that it is available when the actual load (register operation) is executed.

c) performing at least said prefetch operation out-of-order with respect to said preceding instruction to prefetch data, wherein performing said prefetch operation comprises:

c1) calculating a speculative target memory address utilizing contents of at least one register without regard for whether said contents will be modified between calculation of said speculative target memory address and performing said register operation. It should be noted from Fig.2(a) that the target prefetch addresses are calculated using at least the prev_addr field of the RPT entry. However, the data stored in the prev_addr field within the RPT comes from at least one register. For instance, looking at Fig.1, load instructions which reference registers appear at addresses 500, 504, and 512. Taking the load at address 500, 0 would be added to the value in r2 (as is known in the art) and data would be loaded from the resulting address into r4. In this case, the value in r2 is 50000, as shown in the prev_addr RPT field in Fig.3(b). The prefetch addresses are all built of the prev_addr field, as shown in Fig.2, and so the contents of the register are utilized in calculating a speculative target memory address. Finally, the examiner has found no portion of Chen which suggests calculating a speculative target memory address utilizing contents of at least one register **with regard** for whether said contents will be modified

between calculation of said speculative target memory address and performing said register operation. Therefore, the examiner concludes that Chen has taught calculating a speculative target memory address utilizing contents of at least one register **without regard** for whether said contents will be modified between calculation of said speculative target memory address and performing said register operation, which meets applicant's claim limitation.

c2) thereafter initiating a fetch of said data from a memory location associated within said speculative target memory address. Clearly, since the operation is a prefetch, data will be fetched from the speculative prefetch address.

d) thereafter, separately executing said register operation to place said data into a register among said plurality of registers specified by said load instruction. The reason that data is prefetched is so that it is available when the actual load (register operation) is executed.

12. Referring to claim 19, Chen has taught a processor as described in claim 17. Furthermore, it is inherent that said execution circuitry executes said register operation in-order with respect to said preceding instruction. More specifically, if an instruction is preceding said register operation, then it will be executed before said register operation. Note, that the examiner is interpreting preceding instruction as being preceding in terms of execution, not actual program order.

13. Referring to claim 23, Chen has taught a processor as described in claim 17. Chen has further taught that said execution circuitry stores said data prefetched in response to said prefetch operation in a temporary register. See the abstract and Fig.4. Note that data is prefetched into a

data cache, which is a temporary register. That is, it is temporary storage which holds the most recently accessed data.

14. Referring to claim 25, Chen has taught a method as described in claim 18. Furthermore, it is inherent that the method further comprises executing said register operation in-order with respect to said preceding instruction. More specifically, if an instruction is preceding said register operation, then it will be executed before said register operation. Note, that the examiner is interpreting preceding instruction as being preceding in terms of execution, not actual program order.

15. Referring to claim 29, Chen has taught a method as described in claim 18. Chen has further taught that performing said prefetch operation comprises storing said data in a temporary register. See the abstract and Fig.4. Note that data is prefetched into a data cache, which is a temporary register. That is, it is temporary storage which holds the most recently accessed data.

Claim Rejections - 35 USC § 103

16. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

17. Claims 20 and 26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chen, as applied above.

18. Referring to claim 20, Chen has taught a processor as described in claim 17, Chen has not explicitly taught that said execution circuitry executes said register operation out of order with

respect to said preceding instruction. However, Official Notice is taken that out of order execution and its advantages are well known and accepted in the art. More specifically, executing instructions out of order allows a system to execute instructions that are ready now (say the register operation) instead of stalling until a dependent instruction can be executed (say the preceding instruction). This increases efficiency over plain in-order executing machines. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Chen to execute the register operation out of order with respect to said preceding instruction.

19. Referring to claim 26, Chen has taught a method as described in claim 18. Chen has not explicitly taught executing said register operation out of order with respect to said preceding instruction. However, Official Notice is taken that out of order execution and its advantages are well known and accepted in the art. More specifically, executing instructions out of order allows a system to execute instructions that are ready now (say the register operation) instead of stalling until a dependent instruction can be executed (say the preceding instruction). This increases efficiency over plain in-order executing machines. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Chen to execute the register operation out of order with respect to said preceding instruction.

20. Claims 21 and 27 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chen, as applied above, in view of DeLano et al., U.S. Patent No. 5,396,604 (as applied in the previous Office Action and herein referred to as DeLano).

21. Referring to claim 21, Chen has taught a processor as described in claim 17. Chen has not explicitly taught that said prefetch operation and said register operation have a same operation code (opcode). However, DeLano has taught the concept of a prefetch and corresponding register operation having the same opcode. See column 5, lines 20-24, and column 2, lines 48-50. It should be realized that a prefetch is implemented in the same fashion as the actual register operation (as a load instruction with a single opcode). The load is viewed by the system as a prefetch instruction only when register 0 is specified within the instruction. A person of ordinary skill in the art would have recognized that by being able to specify multiple instructions with a single opcode, the overall number of opcodes used within the system would be reduced by 1, which would possibly result in the reduction of the amount of bits required to encode an instruction. For instance, if there were 8 instructions (not including the prefetch instruction), then the 8 instructions would be represented by 3-bit opcodes (000-111). Adding a prefetch operation with a different opcode would require 4-bit opcodes (0000-1000). By using an already existing opcode for the prefetch opcode the programmer would be able to keep the amount of opcode bits at a minimum. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Chen such that said prefetch operation and said register operation have a same operation code, as taught by DeLano.

22. Referring to claim 27, Chen has taught a method as described in claim 18. Chen has not explicitly taught translating said load instruction into prefetch register operations having a same operation code (opcode). However, DeLano has taught the concept of a prefetch and corresponding register operation having the same opcode. See column 5, lines 20-24, and column 2, lines 48-50. It should be realized that a prefetch is implemented in the same fashion

as the actual register operation (as a load instruction with a single opcode). The load is viewed by the system as a prefetch instruction only when register 0 is specified within the instruction. A person of ordinary skill in the art would have recognized that by being able to specify multiple instructions with a single opcode, the overall number of opcodes used within the system would be reduced by 1, which would possibly result in the reduction of the amount of bits required to encode an instruction. For instance, if there were 8 instructions (not including the prefetch instruction), then the 8 instructions would be represented by 3-bit opcodes (000-111). Adding a prefetch operation with a different opcode would require 4-bit opcodes (0000-1000). By using an already existing opcode for the prefetch opcode the programmer would be able to keep the amount of opcode bits at a minimum. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Chen such that said prefetch operation and said register operation have a same operation code, as taught by DeLano.

23. Claims 24 and 30 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chen, as applied above, in view of Konigsburg et al., U. S. Patent No. 5,931,957 (as applied in the previous Office Action and herein referred to as Konigsburg).

24. Referring to claim 24, Chen has taught a processor as described in claim 17. Chen has not explicitly taught a data hazard detector that, in response to detection of a hazard for said data, signals said processor to discard said data and said register operation. However, Konigsburg has taught that speculative instructions along a mispredicted branch path must be flushed and their associated results discarded since any changes made to data by these instructions would cause a data hazard in that the data modified by these instructions should not be accessed by subsequent

instructions. See column 6, line 66, to column 7, line 7. This type of flushing and discarding is well known and expected in the art and a person of ordinary skill in the art would have recognized that if a prefetch instruction and the associated load instruction are along a mispredicted branch path, then the data loaded by the instructions is undesired, and therefore, should be discarded. Also, the instructions should be cancelled so that they cannot complete and make any undesired changes to the system. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to implement, within Chen's system, a data hazard detector that, in response to detection of a hazard for said data, signals said processor to discard said data and said register operation.

25. Referring to claim 30, Chen has taught a method as described in claim 18. Chen has not explicitly taught a data hazard detector that, in response to detection of a hazard for said data, signals said processor to discard said data and said register operation. However, Konigsburg has taught that speculative instructions along a mispredicted branch path must be flushed and their associated results discarded since any changes made to data by these instructions would cause a data hazard in that the data modified by these instructions should not be accessed by subsequent instructions. See column 6, line 66, to column 7, line 7. This type of flushing and discarding is well known and expected in the art and a person of ordinary skill in the art would have recognized that if a prefetch instruction and the associated load instruction are along a mispredicted branch path, then the data loaded by the instructions is undesired, and therefore, should be discarded. Also, the instructions should be cancelled so that they cannot complete and make any undesired changes to the system. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to implement, within Chen's system, a data

hazard detector that, in response to detection of a hazard for said data, signals said processor to discard said data and said register operation.

Allowable Subject Matter

26. Claims 22 and 28 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

Conclusion

27. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. Applicant is reminded that in amending in response to a rejection of claims, the patentable novelty must be clearly shown in view of the state of the art disclosed by the references cited and the objections made. Applicant must also show how the amendments avoid such references and objections. See 37 CFR § 1.111(c).

IBM Technical Disclosure Bulletin, Vol.38, No.06, June 1995, "Methods of Specifying Data Prefetching without using a Separate Instruction," has mentioned a touch instruction which is used to touch the data cache (prefetch data into the cache), so data will be ready for subsequent load instructions.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to David J. Huisman whose telephone number is (703) 305-7811. The examiner can normally be reached on Monday-Friday (8:00-4:30).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (703) 305-9712. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

DJH
David J. Huisman
July 7, 2004

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100